L Number	Hits	Search Text	DB	Time stamp
2	405	712/225.ccls.	USPAT; US-PGPUB	2004/05/27 11:40
3	125271	(memory cache).ab.	USPAT; US-PGPUB	2004/05/27 13:19
4	157531	(load or move or (data adj transfer)).ab.	USPAT; US-PGPUB	2004/05/27 13:19
5	3873	(stall latency).ab.	USPAT; US-PGPUB	2004/05/27 13:19
6	7173	prefetch$4 or pre-fetch$4	USPAT; US-PGPUB	2004/05/27 13:20
7	138	((memory cache).ab.) and ((load or move or (data adj transfer)).ab.) and ((stall latency).ab.)	USPAT; US-PGPUB	2004/05/27 12:54
8	51	((load or move or (data adj transfer)).ab.) and ((stall latency).ab.) and search$4	USPAT; US-PGPUB	2004/05/27 12:58
9	28	((load or move or (data adj transfer)).ab.) and ((stall latency).ab.) and scan$5	USPAT; US-PGPUB	2004/05/27 12:58
10	213	((memory cache).ab.) and ((stall latency).ab.) and (prefetch$4 or pre-fetch$4)	USPAT; US-PGPUB	2004/05/27 13:14
11	102	(GUPTA-RAJIV GUPTA-RAJIV-K KARP-ALAN KARP-ALAN-H KARP-ALAN-HERSH).in.	USPAT; US-PGPUB	2004/05/27 13:19
12	854309	memory cache	EPO; JPO; DERWENT; IBM_TDB	2004/05/27 13:19
13	1387799	(load or move or (data adj transfer))	EPO; JPO; DERWENT; IBM_TDB	2004/05/27 13:19
14	12801	stall latency	EPO; JPO; DERWENT; IBM_TDB	2004/05/27 13:19
15	2834	prefetch$4 or pre-fetch$4	EPO; JPO; DERWENT; IBM_TDB	2004/05/27 13:20
31	63	(decoder with (search$4 scan$5) with load)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/27 13:26
38	8067	(search$4 scan$5) with (instruction)	USPAT; US-PGPUB	2004/05/27 13:57
39	298	((search$4 scan$5) with (instruction)) with load	USPAT; US-PGPUB	2004/05/27 13:57
40	7	(((search$4 scan$5) with (instruction)) with load) with (prefetch$4 or pre-fetch$4)	USPAT; US-PGPUB	2004/05/27 14:00
41	1	5,377,336.pn.	USPAT; US-PGPUB	2004/05/27 14:01
54	52	5377336.uref.	USPAT	2004/05/27 14:17
16	412	(memory cache) and ((load or move or (data adj transfer))) and (stall latency)	EPO; JPO; DERWENT; IBM_TDB	2004/05/27 15:16

Your search matched **1** of **1040503** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

(effective hardware based data prefetching<in>ti) Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **Effective hardware-based data prefetching for high-performance processors**
Tien-Fu Chen; Jean-Loup Baer;
Computers, IEEE Transactions on , Volume: 44 , Issue: 5 , May 1995
Pages:609 - 623

[Abstract] [PDF Full-Text (1408 KB)] **IEEE JNL**

Terms used **hardware prefetch** Found **22,345** of **134,837**

Sort results by relevance Save results to a Binder Try an Advanced Search

Display results expanded form Search Tips Try this search in The ACM Guide

Open results in a new window

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 A performance study of software and hardware data prefetching schemes

T.-F. Chen, J.-L. Baer

April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture**, Volume 22 Issue 2

Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, citings, index terms

Prefetching, i.e., exploiting the overlap of processor computations with data accesses, is one of several approaches for tolerating memory latencies. Prefetching can be either hardware-based or software-directed or a combination of both. Hardware-based prefetching, requiring some support unit connected to the cache, can dynamically handle prefetches at run-time without compiler intervention. Software-directed approaches rely on compiler technology to insert explicit prefetch instructions. Mowry ...

2 Guided region prefetching: a cooperative hardware/software approach

Zhenlin Wang, Doug Burger, Kathryn S. McKinley, Steven K. Reinhardt, Charles C. Weems

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2

Full text available: pdf(171.47 KB) Additional Information: full citation, abstract, references

Despite large caches, main-memory access latencies still cause significant performance losses in many applications. Numerous hardware and software prefetching schemes have been proposed to tolerate these latencies. Software prefetching typically provides better prefetch accuracy than hardware, but is limited by prefetch instruction overheads and the compiler's limited ability to schedule prefetches sufficiently far in advance to cover level-two cache miss latencies. Hardware prefetching can be e ...

3 Session 17: architecture: Sunder: a programmable hardware prefetch architecture for numerical loops

Tzi-cker Chiueh

November 1994 **Proceedings of the 1994 ACM/IEEE conference on Supercomputing**

Full text available: pdf(922.38 KB) Additional Information: full citation, abstract, references, citings

Beyond data caching, data prefetching is by far the most effective way to address the memory access bottleneck associated with high-performance processors. This is particularly true for scientific programs whose working sets cannot be easily fit into the on-chip data cache. This paper proposes a new data prefetching architecture called **Sunder,** *which combines the flexibility and accurateness of software prefetching and the transparency and low-overhead of hardware prefetching. Th ...*

4 Cooperative prefetching: compiler and hardware support for effective instruction prefetching in modern processors

Chi-Keung Luk, Todd C. Mowry
November 1998 **Proceedings of the 31st annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(3.41 MB) Additional Information: full citation, references, citings, index terms

5 Tango: a hardware-based data prefetching technique for superscalar processors
Shlomit S. Pinter, Adi Yoaz
December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(1.46 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

We present a new hardware-based data prefetching mechanism for enhancing instruction level parallelism and improving the performance of superscalar processors. The emphasis in our scheme is on the effective utilization of slack time and hardware resources not used for the main computation. The scheme suggests a new hardware construct, the program progress graph (PPG), as a simple extension to the branch target buffer (BTB). We use the PPG for implementing a fast pre-program counter pre-PC, that ...

Keywords: LRU mechanism, SPEC92 benchmark, Tango, base line architecture, branch target buffer, hardware resources, hardware-based data prefetching technique, instruction level parallelism, instruction prefetching, memory reference instructions, parallel processing, performance, program progress graph, simulation results, slack time, superscalar processors

6 Hardware-only stream prefetching and dynamic access ordering
Chengqiang Zhang, Sally A. McKee
May 2000 **Proceedings of the 14th international conference on Supercomputing**
Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, citings, index terms

Memory system bottlenecks limit performance for many applications, and computations with strided access patterns are among the hardest hit. The streams used in such applications have extremely poor cache behavior. These access patterns have the advantage of being predictable, though, and this can be exploited to improve the efficiency of the memory subsystem in two ways: memory latencies can be masked by prefetching stream data, and the latencies can be reduced by reordering stream accesses ...

7 Hardware-driven prefetching for pointer data references
Chi-Hung Chi, Chin-Ming Cheung
July 1998 **Proceedings of the 12th international conference on Supercomputing**
Full text available: pdf(1.06 MB) Additional Information: full citation, references, citings, index terms

8 Data prefetch mechanisms
Steven P. Vanderwiel, David J. Lilja
June 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 2
Full text available: pdf(172.07 KB) Additional Information: full citation, abstract, references, citings, index terms, review

The expanding gap between microprocessor and DRAM performance has necessitated the use of increasingly aggressive techniques designed to reduce or hide the latency of main memory access. Although large cache hierarchies have proven to be effective in reducing this latency for the most frequently used data, it is still not uncommon for many programs to spend more than half their run times stalled on memory requests. Data prefetching has been proposed as a technique for hiding the access lat ...

Keywords: memory latency, prefetching

9 Effective jump-pointer prefetching for linked data structures

Amir Roth, Gurindar S. Sohi

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture**, Volume 27 Issue 2

Full text available: pdf(113.33 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Current techniques for prefetching linked data structures (LDS) exploit the work available in one loop iteration or recursive call to overlap pointer chasing latency. Jump pointers, which provide direct access to non-adjacent nodes, can be used for prefetching when loop and recursive procedure bodies are small and do not have sufficient work to overlap a long latency. This paper describes a framework for jump-pointer prefetching (JPP) that supports four prefetching idioms: queue, full, chain, an ...

10 A general framework for prefetch scheduling in linked data structures and its application to multi-chain prefetching

Seungryul Choi, Nicholas Kohout, Sumit Pamnani, Dongkeun Kim, Donald Yeung

May 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 2

Full text available: pdf(2.45 MB) Additional Information: full citation, abstract, references, index terms

Pointer-chasing applications tend to traverse composite data structures consisting of multiple independent pointer chains. While the traversal of any single pointer chain leads to the serialization of memory operations, the traversal of independent pointer chains provides a source of memory parallelism. This article investigates exploiting such *interchain memory parallelism* for the purpose of memory latency tolerance, using a technique called *multi--chain prefetching*. Previous work ...

Keywords: Data prefetching, memory parallelism, pointer-chasing code

11 Software prefetching

David Callahan, Ken Kennedy, Allan Porterfield

April 1991 **Proceedings of the fourth international conference on Architectural support for programming languages and operating systems**, Volume 19 , 25 , 26 Issue 2 , Special Issue , 4

Full text available: pdf(1.25 MB) Additional Information: full citation, references, citings, index terms

12 Architectural and compiler support for effective instruction prefetching: a cooperative approach

February 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 1

Full text available: pdf(432.96 KB) Additional Information: full citation, abstract, references, index terms, review

Instruction cache miss latency is becoming an increasingly important performance bottleneck, especially for commercial applications. Although instruction prefetching is an attractive technique for tolerating this latency, we find that existing prefetching schemes are insufficient for modern superscalar processors, since they fail to issue prefetches early enough (particularly for nonsequential accesses). To overcome these limitations, we propose a new instruction prefetching technique where ...

Keywords: compiler optimization, instruction prefetching

13 Speculative precomputation: long-range prefetching of delinquent loads

Jamison D. Collins, Hong Wang, Dean M. Tullsen, Christopher Hughes, Yong-Fong Lee, Dan Lavery, John P. Shen

May 2001 **ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual international symposium on Computer architecture**, Volume 29 Issue 2

Full text available: pdf(995.50 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

This paper explores Speculative Precomputation, a technique that uses idle thread context in a multithreaded architecture to improve performance of single-threaded applications. It attacks program stalls from data cache misses by pre-computing future memory accesses in available thread contexts, and prefetching these data. This technique is evaluated by simulating the performance of a research processor based on the Itanium™ ISA supporting Simultaneous Multithreading. Two primary for ...

14 Call graph prefetching for database applications

Murali Annavaram, Jignesh M. Patel, Edward S. Davidson

November 2003 **ACM Transactions on Computer Systems (TOCS)**, Volume 21 Issue 4

Full text available: pdf(701.71 KB) Additional Information: full citation, abstract, references, index terms

With the continuing technological trend of ever cheaper and larger memory, most data sets in database servers will soon be able to reside in main memory. In this configuration, the performance bottleneck is likely to be the gap between the processing speed of the CPU and the memory access latency. Previous work has shown that database applications have large instruction and data footprints and hence do not use processor caches effectively. In this paper, we propose Call Graph Prefetching (CGP), ...

Keywords: Instruction cache prefetching, call graph, database

15 Speeding up irregular applications in shared-memory multiprocessors: memory binding and group prefetching

Zheng Zhang, Josep Torrellas

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2

Full text available: pdf(1.74 MB) Additional Information: full citation, abstract, references, citings, index terms

While many parallel applications exhibit good spatial locality, other important codes in areas like graph problem-solving or CAD do not. Often, these irregular codes contain small records accessed via pointers. Consequently, while the former applications benefit from long cache lines, the latter prefer short lines. One good solution is to combine short lines with prefetching. In this way, each application can exploit the amount of spatial locality that it has. However, prefetching, if provided, ...

16 DSTRIDE: data-cache miss-address-based stride prefetching scheme for multimedia processors

Hariprakash. G, Achutharaman. R, Amos R. Omondi

January 2001 **Australian Computer Science Communications , Proceedings of the 6th Australasian conference on Computer systems architecture**, Volume 23 Issue 4

Full text available: pdf(928.14 KB) Publisher Site Additional Information: full citation, abstract, references

Prefetching reduces cache miss latency by moving data up in memory hierarchy before they are actually needed. Recent hardware-based stride prefetching techniques mostly rely on the processor pipeline information (e.g. program counter and branch prediction table) for prediction. Continuing developments in processor microarchitecture drastically change core pipeline design and require that existing hardware-based stride prefetching techniques be adapted to the evolving new processor architectures. ...

17 Speculative prefetching
Y. Jégou, O. Temam
August 1993 **Proceedings of the 7th international conference on Supercomputing**

Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, citings, index terms

A hardware prefetching mechanism named Speculative Prefetching is proposed. This scheme detects vector accesses issued by a load/store instruction and prefetches the corresponding data. The scheme requires no software add-on, and in some cases it is more powerful than software techniques for identifying regular accesses. The tradeoffs related to its hardware implementation are extensively discussed in order to finely tune the mechanism. Experiments show that average memory ...

18 The interaction of software prefetching with ILP processors in shared-memory systems
Parthasarathy Ranganathan, Vijay S. Pai, Hazim Abdel-Shafi, Sarita V. Adve
May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture**, Volume 25 Issue 2

Full text available: pdf(2.44 MB) Additional Information: full citation, abstract, references, citings, index terms

Current microprocessors aggressively exploit instruction-level parallelism (ILP) through techniques such as multiple issue, dynamic scheduling, and non-blocking reads. Recent work has shown that memory latency remains a significant performance bottleneck for shared-memory multiprocessor systems built of such processors.This paper provides the first study of the effectiveness of software-controlled non-binding prefetching in shared memory multiprocessors built of state-of-the-art ILP-based proces ...

19 Cooperative shared memory: software and hardware for scalable multiprocessor
Mark D. Hill, James R. Larus, Steven K. Reinhardt, David A. Wood
September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems**, Volume 27 Issue 9

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

We believe the absence of massively-parallel, shared-memory machines follows from the lack of a shared-memory programming performance model that can inform programmers of the cost of operations (so they can avoid expensive ones) and can tell hardware designers which cases are common (so they can build simple hardware to optimize them). Cooperative shared memory, our approach to shared-memory design, addresses this problem. Our initial implementation of cooperativ ...

20 Cooperative shared memory: software and hardware for scalable multiprocessors
Mark D. Hill, James R. Larus, Steven K. Reinhardt, David A. Wood
November 1993 **ACM Transactions on Computer Systems (TOCS)**, Volume 11 Issue 4

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

We believe the paucity of massively parallel, shared-memory machines follows from the lack of a shared-memory programming performance model that can inform programmers of the cost of operations (so they can avoid expensive ones) and can tell hardware designers which cases are common (so they can build simple hardware to optimize them). Cooperative shared memory, our approach to shared-memory design, addresses this problem. Our initial implementation of cooperative shared memory u ...

Keywords: cache coherence, directory protocols, memory systems, programming model, shared-memory multiprocessors

Terms used
An effective on chip preloading scheme to reduce data access penalty

Found **64,680** of **134,837**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown
Relevance scale

1 An effective on-chip preloading scheme to reduce data access penalty
Jean-Loup Baer, Tien-Fu Chen
August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing**
Full text available: pdf(1.18 MB) Additional Information: full citation, references, citings, index terms

2 Tango: a hardware-based data prefetching technique for superscalar processors
Shlomit S. Pinter, Adi Yoaz
December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(1.46 MB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

We present a new hardware-based data prefetching mechanism for enhancing instruction level parallelism and improving the performance of superscalar processors. The emphasis in our scheme is on the effective utilization of slack time and hardware resources not used for the main computation. The scheme suggests a new hardware construct, the program progress graph (PPG), as a simple extension to the branch target buffer (BTB). We use the PPG for implementing a fast pre-program counter pre-PC, that ...

Keywords: LRU mechanism, SPEC92 benchmark, Tango, base line architecture, branch target buffer, hardware resources, hardware-based data prefetching technique, instruction level parallelism, instruction prefetching, memory reference instructions, parallel processing, performance, program progress graph, simulation results, slack time, superscalar processors

3 An efficient architecture for loop based data preloading
William Y. Chen, Roger A. Bringmann, Scott A. Mahlke, Richard E. Hank, James E. Sicolo
December 1992 **ACM SIGMICRO Newsletter , Proceedings of the 25th annual international symposium on Microarchitecture**, Volume 23 Issue 1-2
Full text available: pdf(1.65 MB) Additional Information: full citation, references, citings, index terms

4 A performance study of software and hardware data prefetching schemes
T.-F. Chen, J.-L. Baer
April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture**, Volume 22 Issue 2

Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, citings, index terms

Prefetching, i.e., exploiting the overlap of processor computations with data accesses, is one of several approaches for tolerating memory latencies. Prefetching can be either hardware-based or software-directed or a combination of both. Hardware-based prefetching, requiring some support unit connected to the cache, can dynamically handle prefetches at run-time without compiler intervention. Software-directed approaches rely on compiler technology to insert explicit prefetch instructions. Mowry ...

5 An effective programmable prefetch engine for on-chip caches

Tien-Fu Chen

December 1995 **Proceedings of the 28th annual international symposium on Microarchitecture**

Full text available: pdf(721.83 KB) Additional Information: full citation, references, citings, index terms

6 Reducing memory latency via non-blocking and prefetching caches

Tien-Fu Chen, Jean-Loup Baer

September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems**, Volume 27 Issue 9

Full text available: pdf(1.36 MB) Additional Information: full citation, references, citings, index terms

7 Data prefetch mechanisms

Steven P. Vanderwiel, David J. Lilja

June 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 2

Full text available: pdf(172.07 KB) Additional Information: full citation, abstract, references, citings, index terms, review

The expanding gap between microprocessor and DRAM performance has necessitated the use of increasingly aggressive techniques designed to reduce or hide the latency of main memory access. Although large cache hierarchies have proven to be effective in reducing this latency for the most frequently used data, it is still not uncommon for many programs to spend more than half their run times stalled on memory requests. Data prefetching has been proposed as a technique for hiding the access lat ...

Keywords: memory latency, prefetching

8 DSTRIDE: data-cache miss-address-based stride prefetching scheme for multimedia processors

Hariprakash. G, Achutharaman. R, Amos R. Omondi

January 2001 **Australian Computer Science Communications , Proceedings of the 6th Australasian conference on Computer systems architecture**, Volume 23 Issue 4

Full text available: pdf(928.14 KB) Publisher Site Additional Information: full citation, abstract, references

Prefetching reduces cache miss latency by moving data up in memory hierarchy before they are actually needed. Recent hardware-based stride prefetching techniques mostly rely on the processor pipeline information (e.g. program counter and branch prediction table) for prediction. Continuing developments in processor microarchitecture drastically change core pipeline design and require that existing hardware-based stride prefetching techniques be adapted to the evolving new processor architectures. ...

9 Utilizing reuse information in data cache management

Jude A. Rivers, Edward S. Tam, Gary S. Tyson, Edward S. Davidson, Matt Farrens
July 1998 **Proceedings of the 12th international conference on Supercomputing**
Full text available: pdf(1.12 MB) Additional Information: full citation, references, citings, index terms

Keywords: effective address, multi-lateral caches, program counter

10 Design and evaluation of a compiler algorithm for prefetching
Todd C. Mowry, Monica S. Lam, Anoop Gupta
September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems**, Volume 27 Issue 9
Full text available: pdf(1.70 MB) Additional Information: full citation, references, citings, index terms

11 Pseudo vector processor based on register-windowed superscalar pipeline
K. Nakazawa, H. Nakamura, H. Imori, S. Kawabe
December 1992 **Proceedings of the 1992 ACM/IEEE conference on Supercomputing**
Full text available: pdf(1.19 MB) Additional Information: full citation, references, citings, index terms

12 Zero-cycle loads: microarchitecture support for reducing load latency
Todd M. Austin, Gurindar S. Sohi
December 1995 **Proceedings of the 28th annual international symposium on Microarchitecture**
Full text available: pdf(1.35 MB) Additional Information: full citation, references, citings, index terms

13 Correlated load-address predictors
Michael Bekerman, Stephan Jourdan, Ronny Ronen, Gilad Kirshenboim, Lihu Rappoport, Adi Yoaz, Uri Weiser
May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture**, Volume 27 Issue 2
Full text available: pdf(149.18 KB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

As microprocessors become faster, the relative performance cost of memory accesses increases. Bigger and faster caches significantly reduce the absolute load-to-use time delay. However, increase in processor operational frequencies impairs the relative load-to-use latency, measured in processor cycles (e.g. from two cycles on the Pentium® processor to three cycles or more in current designs). Load-address prediction techniques were introduced to partially cut the load-to-use latency. Thi ...

Keywords: context-based predictor, global correlation, load-address prediction, predictor implementation, recursive data structures

14 Run-time adaptive cache hierarchy management via reference analysis
Teresa L. Johnson, Wen-mei W. Hwu
May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture**, Volume 25 Issue 2
Full text available: pdf(2.09 MB) Additional Information: full citation, abstract, references, citings, index terms

Improvements in main memory speeds have not kept pace with increasing processor clock

frequency and improved exploitation of instruction-level parallelism. Consequently, the gap between processor and main memory performance is expected to grow, increasing the number of execution cycles spent waiting for memory accesses to complete. One solution to this growing problem is to reduce the number of cache misses by increasing the effectiveness of the cache hierarchy. In this paper we present a techni ...

15 Improving data cache performance by pre-executing instructions under a cache miss
James Dundas, Trevor Mudge
July 1997 **Proceedings of the 11th international conference on Supercomputing**
Full text available: pdf(1.04 MB) Additional Information: full citation, references, citings, index terms

16 Compiler-based prefetching for recursive data structures
Chi-Keung Luk, Todd C. Mowry
October 1996 **Proceedings of the seventh international conference on Architectural support for programming languages and operating systems**, Volume 30 , 31 Issue 5 , 9
Full text available: pdf(1.51 MB) Additional Information: full citation, abstract, references, citings, index terms

Software-controlled data prefetching offers the potential for bridging the ever-increasing speed gap between the memory subsystem and today's high-performance processors. While prefetching has enjoyed considerable success in array-based numeric codes, its potential in pointer-based applications has remained largely unexplored. This paper investigates compiler-based prefetching for pointer-based applications---in particular, those containing recursive data structures. We identify the fundamental ...

17 An architecture for software-controlled data prefetching
Alexander C. Klaiber, Henry M. Levy
April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture**, Volume 19 Issue 3
Full text available: pdf(1.16 MB) Additional Information: full citation, references, citings, index terms

18 Evaluating stream buffers as a secondary cache replacement
S. Palacharla, R. E. Kessler
April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture**, Volume 22 Issue 2
Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings, index terms

Today's commodity microprocessors require a low latency memory system to achieve high sustained performance. The conventional high-performance memory system provides fast data access via a large secondary cache. But large secondary caches can be expensive, particularly in large-scale parallel systems with many processors (and thus many caches).We evaluate a memory system design that can be both cost-effective as well as provide better performance, particularly for scientific workloads: a single ...

19 Dependence based prefetching for linked data structures
Amir Roth, Andreas Moshovos, Gurindar S. Sohi
October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 32 , 33 Issue 5 , 11
Full text available: pdf(1.81 MB) Additional Information: full citation, abstract, references, citings, index terms

We introduce a dynamic scheme that captures the accesspat-terns of linked data structures and can be used to predict future accesses with high accuracy. Our technique exploits the dependence relationships that exist between loads that produce addresses and loads that

consume these addresses. By identzj+ing producer-consumer pairs, we construct a compact internal representation for the associated structure and its traversal. To achieve a prefetching eflect, a small prefetch engine speculatively t ...

20 Tolerating latency in multiprocessors through compiler-inserted prefetching

Todd C. Mowry

February 1998 **ACM Transactions on Computer Systems (TOCS)**, Volume 16 Issue 1

Full text available: pdf(410.70 KB) Additional Information: full citation, abstract, references, citings, index terms, review

The large latency of memory accesses in large-scale shared-memory multiprocessors is a key obstacle to achieving high processor utilization. Software-controlled prefetching is a technique for tolerating memory latency by explicitly executing instructions to move data close to the processor before the data are actually needed. To minimize the burden on the programmer, compiler support is needed to automatically insert prefetch instructions into the code. A key challenge when ...

Keywords: compiler optimization, prefetching